

My BASIC programs

TO FIND HIGHEST AND LOWEST OF AN ARRAY

$A(\cdot)$ is array being searched
 N is number of elements in array

```
H=0 \ L=0
FOR I=1 TO N
IF H<A(I) THEN H=A(I)
IF L>A(I) THEN L=A(I)
NEXT
```

H will equal highest element
 L will equal lowest element

TO SCALE A ONE-DIMENSIONAL ARRAY FOR PLOTTING AND PLOT IT

• AS A BAR GRAPH •

$Y1$ is bottom of plotting area ($0 \leq Y1 \leq 47$)
 $X1$ is left side of plotting area ($0 \leq X1 < X2$)
 $X2$ is right side of plotting area ($X1 < X2 \leq 127$)



```
IF (X2-X1) < N THEN PRINT "ARRAY TOO LARGE FOR PLOTTING AREA" \ GOTO ***
```

```
GOSUB find-highest-routine \ GOSUB +++
```

```
S = H / 47 \ I = 0
```

```
FOR X = X1 TO X2 STEP (X2-X1) / N
```

```
I = I + 1 \ IF I > N THEN ***
```

```
FOR Y = Y1 TO A(I) / S
```

```
PLOT X, Y, 1
```

```
NEXT \ NEXT
```

```
*** end
```

N must be less than or equal to $(X2-X1)$
 $\leq$

S will equal scaling factor

```
+++ FOR X = X1 TO X2 \ PLOT X, Y1, 1 \ PLOT X, 47, 1 \ NEXT
FOR Y = Y1 TO 47 \ PLOT X1, Y, 1 \ PLOT X2, Y, 1 \ NEXT } plot borders
```


TIME routine

legal $\rightarrow H < 24, m < 60, s < 60$

$H2, m2, s2$ = current Hours, Minutes, Seconds already

input when $t = \text{TIME}(\emptyset) \setminus \text{POKE } 3074, \emptyset \setminus \text{POKE } 3075, \emptyset$ was executed.

Figure time to be added in

10 $H1 = \text{TIME}(1) + \text{PEEK}(3074) * 65536 + \text{PEEK}(3075) * 16777216 \setminus H = \text{INT}(H1)$

20 $M1 = (H1 - H) * 60 \setminus m = \text{INT}(M1)$

30 $S1 = (M1 - m) * 60 \setminus S = \text{INT}(S1)$

adjust 40 $S = S + S2 \setminus \text{IF } S > 59 \text{ THEN } M1 = \text{INT}(S/60) \setminus m = m + M1 \setminus S = S - M1 * 60$

time for 50 $M = m + M2 \setminus \text{IF } M > 59 \text{ THEN } H1 = \text{INT}(M/60) \setminus H = H + H1 \setminus m = m - H1 * 60$

correct H, m, s 60 $H = H + H2 \setminus \text{IF } H > 23 \text{ THEN } H = H - 24 \setminus D = D + 1 \leftarrow \text{day counter}$

make 70 $H\$ = \text{STR\$}(H) \setminus H\$ = "00" + H\$ (2) \setminus H\$ = H\$ (LEN(H\$) - 1)$

time into 80 $m\$ = \text{STR\$}(m) \setminus m\$ = "00" + m\$ (2) \setminus m\$ = m\$ (LEN(m\$) - 1)$

a string 90 $S\$ = \text{STR\$}(S) \setminus S\$ = "00" + S\$ (2) \setminus S\$ = S\$ (LEN(S\$) - 1)$

make it 1 day 100 $T\$ = H\$ + ":" + m\$ + ":" + S\$$

Return or !T\$ and RETURN or whatever

FIND and REPLACE function

```
200 DEF FNR(A$,B$,C$)
210 P=FNP(A$,B$) \ I=0 ← POS function
220 IF P<>0 THEN I=1 ELSE IF P=0 THEN GOTO 250
230 IF P=1 THEN A$=C$+A$(P+L2) \ GOTO 210
240 A$=A$(1,P-1)+C$+A$(P+L2) \ GOTO 210
250 RETURN I
260 FNEND
```

Replaces B\$ of A\$ with C\$ (all occurrences of B\$). If B\$ cannot be found in A\$ then it returns a 0 else it returns a 1.

EX: if A\$="TRITE", B\$="IT", C\$="U"

then it will return a 1, and A\$ will then = "TRUE"

POS function

(INSTR)

```
100 DEF FNP(A$,B$)
110   L1=LEN(A$)
120   L2=LEN(B$)
130   FOR P=1 TO L1-L2+1
140     IF A$(P,P+L2-1)=B$ THEN RETURN P
150   NEXT P
160 RETURN 0
170 FNEND
```

Returns the position of the 1st character of B\$ in A\$.
if B\$ is not in A\$ then it returns a 0

EX: if A\$ = "TRITE", B\$ = "IT"

it will return a 3

IVTC RPG II Aircraft seating problem

A# = aircraft's name

S = # of seats

R = # of rows with each row having 3 or less seats

E = # of empty nonexistent seats in last row

DIMENSION S(,3) TO LARGEST AMOUNT OF ROWS IN LARGEST AIRCRAFT

```

10 READ A#,S \ DATA "727",160
20 R=INT(S/3)
30 E=S-R*3 DO NOT FORGET TO SET E=3
40 IF E<>0 THEN E=3-E \ R=R+1 \ T=E-1
50 FOR X=1 TO R
70 S(1,X)=X \ S(X,2)=X+R \ S(X,3)=X+R*2-T
80 NEXT X
90 IF E>0 THEN S(R,3)=0
100 IF E>1 THEN S(R,2)=0
110 PRINT A#," airplane seats",S," people.",%#4I
120 FOR X=1 TO R \ PRINT S(X,1),
130 PRINT TAB(20), \ IF S(X,2)<>0 THEN PRINT S(X,2),
140 PRINT TAB(40), \ IF S(X,3)<>0 THEN PRINT S(X,3)
150 NEXT X \ PRINT %#

```

~~T=1-90~~
~~2=1-20~~
~~3=20-30~~
~~10=90-99~~

- INSIDE A PROGRAM 'LIST' -

DEF FNL1(S,T)

S\$ = STR\$(S) - starting line number

E\$ = STR\$(T) - Last line number

OUT Ø, 12

FF

OUT Ø, 127

BKSPC

OUT Ø, 161

LIST

FOR X=1 TO LEN(S\$)

OUT Ø, ASC(S\$(X))

1st line #

NEXT

OUT Ø, 44

comma

FOR X=1 TO LEN(E\$)

OUT Ø, ASC(E\$(X))

End line #

NEXT

OUT Ø, 13

CR

OUT Ø, 166

CONtime

OUT Ø, 13

CR

STOP

!CHR\$(127), CHR\$(24),

erases CONtime line

RETURN Ø

FNEND

DIM A\$(1024) \ Q\$ = CHR\$(255)
EDIT FN

#19004

escape out with CTL-B

19990
20,000

DEF FNE\$(A\$)

Q\$ = CHR\$(255)

!CHR\$(12), \ C=2 \ A\$ = Q\$ + A\$ + CHR\$(0)

20010 !CHR\$(11),

20020 !A\$(1,C-1) + Q\$ + A\$(C), ~~GOTO 20030~~

20030 I = INF(1) \ IF I > 31 THEN 20100

20040 IF I = 4 THEN C = C - 1 \ IF C < 2 THEN C = 2

20050 IF I = 6 THEN C = C + 1 \ IF C > LEN(A\$) THEN C = LEN(A\$)

20060 IF I = 24 THEN A\$ = "" \ GOTO 20000

20070 IF I = 12 THEN !CHR\$(12), \ GOTO 20020

20080 IF I = 13 THEN !CHR\$(12), \ GOTO 20120

20085 IF I = 2 THEN RETURN A\$(2, LEN(A\$) - 1)

20090 IF I < 32 THEN 20010

20100 IF A\$(C-1, C-1) = CHR\$(13) THEN IF I = 127 THEN !CHR\$(12),

20110 IF I = 127 THEN IF C < 3 THEN 20010 ELSE A\$ = A\$(1, C-2) +
A\$(C) \ C = C - 1 \ GOTO 20010

20120 A\$ = A\$(1, C-1) + CHR\$(I) + A\$(C) \ C = C + 1 \ GOTO 20010

20130 FNE\$

20065 IF I = 9 THEN 20120

20105 IF A\$(C-1, C-1) = CHR\$(9) THEN IF I = 127 THEN !CHR\$(12),

on CPI's, carry set if A is less than the CPI #

ADD PLOTTER/MAN to Advertis

CALL MAN

USE (OUT 0,24)'s to move

man move + add in escape key to return to BASIC

will need to MOVE MOVE away from 7000 because MAN is at 7000

Rename Advertis to DEMO

11
63488
960
8

1024
- 64
960

1 -64

4-1 2+1

3+64 put MOVE at 4A00

19072₁₆ = 4A80₁₆ 01 00 FC

21 00 F8

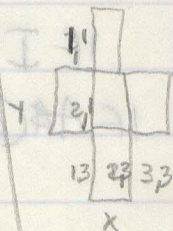
EB

C3 00 01

LHLD XPOS

H=T, L=X

HL=YX



63488
63

63551

4B00-5711

PLOT Via BASIC

~~PLOT IN BASIC~~

$$m = 63488 + \text{INT}((47-y)/3) * 64 + \text{INT}(x/2)$$

$$y1 = (47-y)/3 - \text{INT}$$

PLOT

$$Z1 = \text{INT}((47-y)/3) \setminus Z2 = \text{INT}(x/2)$$

$$m = 63488 + Z1 * 64 + Z2$$

$$Y1 = (47-y)/3 - Z1 \setminus X1 = x/2 - Z2$$

IF Y1 < .3 THEN Y2 = 4 ELSE IF Y1 < .6 THEN Y2 = 2 ELSE Y2 = 1

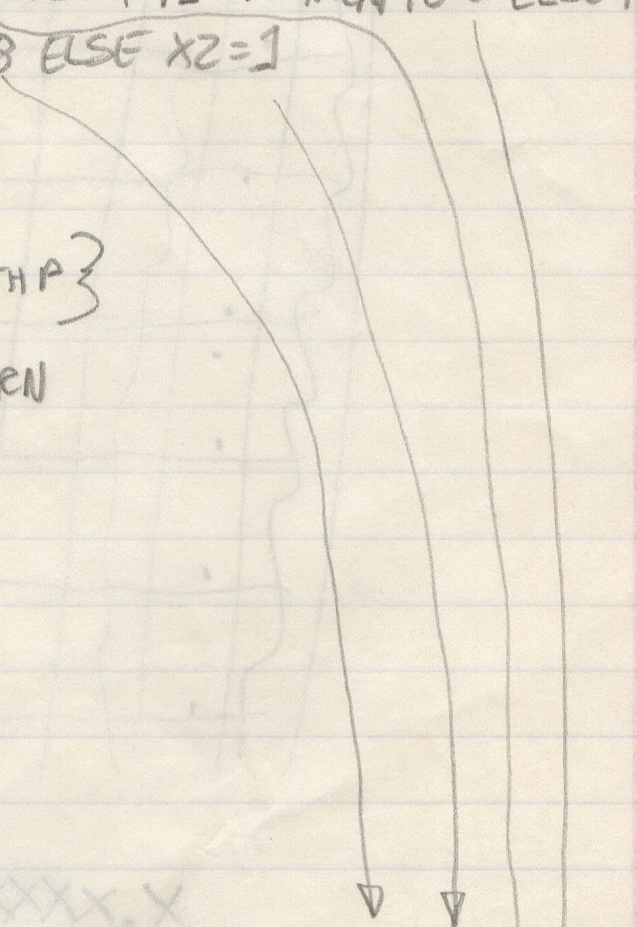
IF X1 < .5 THEN X2 = 8 ELSE X2 = 1

$$A = 63 - (X2 * Y2)$$

$$P = \text{PEEK}(m)$$

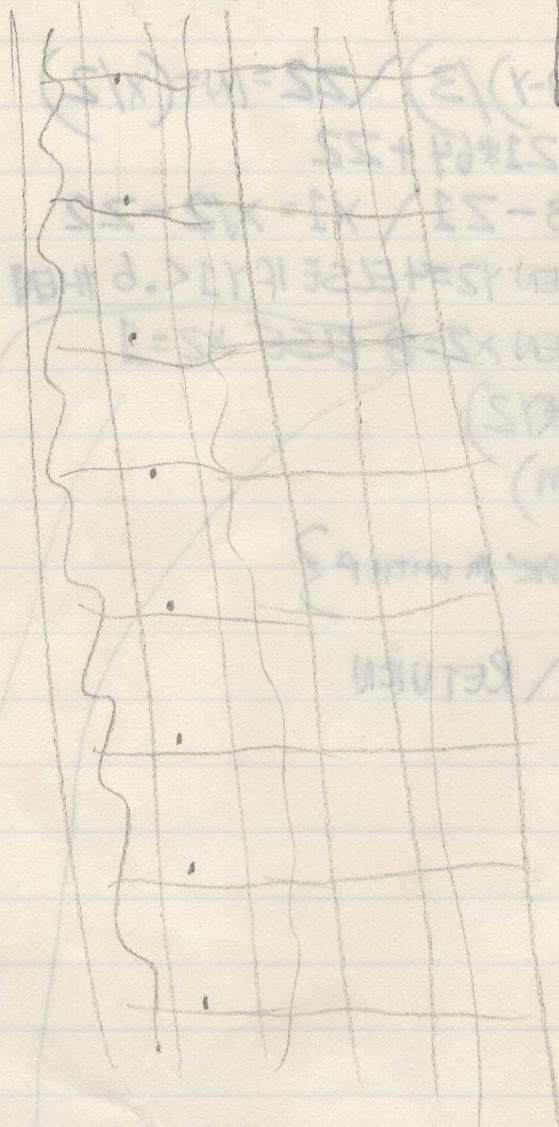
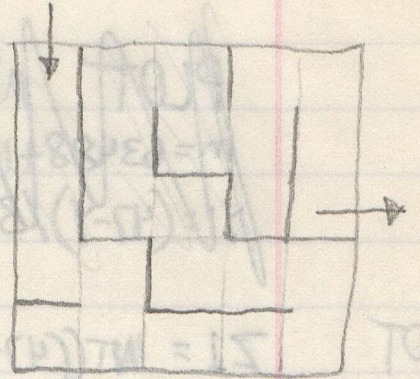
{ NEED TO 'OR' m WITH P }

POKE m, A \ RETURN



32	4
16	2
8	1

Plot V₀ BASIC



X.XXXXXXXE7XX

4	5
5	0
1	8

IF Y DIR = 0, THEN MAKE X DIR 254 or 2

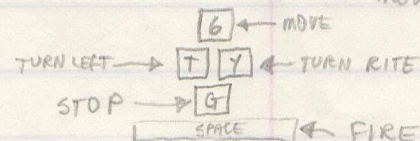
TANK

1 player Testing version

(PLOT SHOT must be loaded too)

```
10 !CHR$(12), \ PLOT 0, 47, 0 \ Z=CALL(18864)
20 X=64 \ Y=24 \ D=1 \ M=1
30 IF INP(0) <> 0 THEN I=INP(1) GOSUB 90
40 IF M < 1 THEN 30 ELSE PLOT X, Y, 0
50 GOSUB 230
60 IF X < 5 THEN X=5 ELSE IF X > 122 THEN X=122
70 IF Y < 5 THEN Y=5 ELSE IF Y > 42 THEN Y=42
80 PLOT X, Y, 1 \ GOTO 30
```

SETUP
X, Y = current position of tank
D = DIRECTION
M = If M=1, THEN TANK IS MOVING



plot new position

*** INPUT HANDLER ***

```
90 IF I=54 THEN M=1 \ GOTO 150
100 IF I=84 THEN D=D+1 \ GOTO 150
110 IF I=89 THEN D=D-1 \ GOTO 150
120 IF I=71 THEN M=0 \ GOTO 150
130 IF I=32 THEN GOSUB 180 \ GOTO 150
140 RETURN
150 IF D > 8 THEN D=D-8 ELSE IF D < 1 THEN D=D+8
160 RETURN
```

MOVE (6)
TURN LEFT (T)
TURN RITE (Y)
STOP (G)
SHOOT (SP)

*** SHOOT ***

only uses ACTIVE1 block

```
180 IF PEEK(19081) = 1 THEN RETURN
190 POKE 19082, X \ POKE 19083, Y \ X1=X \ Y1=Y
200 X=0 \ Y=0 \ GOSUB 230 \ POKE 19084, X \ POKE 19085, Y
210 X=X1 \ Y=Y1 \ POKE 19081, 1 \ RETURN
```

IF X=-1 THEN X=255
IF Y=-1 THEN Y=255

*** Direction to X, Y converter ***

does no overflow checking

```
230 ON D GOTO 240, 250, 260, 270, 280, 290, 300, 310
240 X=X+1 \ RETURN
250 X=X+1 \ Y=Y+1
```

D=1
D=2

4052

260 $Y=Y+1 \setminus \text{RETURN}$

$D=3$

270 $Y=Y+1$

$D=4$

280 $X=X-1 \setminus \text{RETURN}$

$D=5$

290 $X=X-1$

$D=6$

300 $Y=Y-1 \setminus \text{RETURN}$

$D=7$

310 $Y=Y-1 \setminus X=X+1 \setminus \text{RETURN}$

$D=8$


```

10 !CHR$(12),\ PLOT 0,47,0
20 DIM S(46,46)
22 FOR X=0 TO 47
24 PLOT X,0,1 \ PLOT 0,X,1
26 PLOT X,47,1 \ PLOT 47,X,1
28 NEXT
30 PLOT 50,44,0 \ !FREE(0),
40 X1=1 \ Y1=23 \ P1=4
50 X2=46 \ Y2=24 \ P2=2
60 GOTO 210
70 IF INP(0)=0 THEN GOSUB 600 ELSE 75
71 GOSUB 650 (MOVE P1)
72 GOSUB 700 (MOVE P2) (240-310)
73 GOTO 160
75 T$=CHR$(INP(1))
80 IF T$=W THEN Y1=Y1+1 \ P1=1 \ GOSUB 700 (MOVE P2)
90 IF INP(0)=0 Y2=Y2+1 \ P2=1 \ GOSUB 650 (MOVE P1)
100 A X1=X1-1 \ P1=2 \ GOSUB 700 (MOVE P2)
110 O X2=X2-1 \ P2=2 \ GOSUB 650 (MOVE P1)
120 Z Y1=Y1-1 \ P1=3 \ GOSUB 700 (MOVE P2)
130 L Y2=Y2-1 \ P2=3 \ GOSUB 650 (MOVE P1)
140 S X1=X1+1 \ P1=4 \ GOSUB 700 (MOVE P2)
150 P X2=X2+1 \ P2=4 \ GOSUB 650 (MOVE P1)
155 temporary - GOTO 70
160 IF (X1=0) OR (X1=47) OR (Y1=0) OR (Y1=47) THEN 400 (P2 WON)
170 IF (X2=0) OR (X2=47) OR (Y2=0) OR (Y2=47) THEN 500 (P1 WON)
180 IF (X1=X2) OR (Y1=Y2) THEN 500 (TIE)
190 IF S(X1,Y1)=1 THEN 400 (P2 WON)
200 IF S(X2,Y2)=1 THEN 500 (P1 WON)
210 S(X1,Y1)=1 \ S(X2,Y2)=1
220 PLOT X1,Y1,1 \ PLOT X2,Y2,1
230 GOTO 70
300 PLOT 50,24,0 \ "TIE" \ GOTO 999 } TIE
400 PLOT 50,24,0 \ "PLAYER 2 WON" \ GOTO 999 } P2 WON
500 PLOT 50,24,0 \ "PLAYER 1 WON" \ GOTO 999 } P1 WON
600 X=TIME(0) \ PLOT X1,Y1,0 \ PLOT X2,Y2,0
610 IF TIME(0)<15 THEN 610
620 PLOT X1,Y1,1 \ PLOT X2,Y2,1 } OPTIONAL } BLINK
630 IF TIME(0)<30 THEN 630 }
640 RETURN
650 ON P1 GOTO 660,670,680,690 } move P1

```



```

660 Y1=Y1+1 \ GOTO 699
670 X1=X1-1 \ GOTO 699
680 Y1=Y1-1 \ GOTO 699
690 X1=X1+1
699 RETURN
700 ON P2 GOTO 710,720,730,740
710 Y2=Y2+1 \ GOTO 749
720 X2=X2-1 \ GOTO 749
730 Y2=Y2-1 \ GOTO 749
740 X2=X2+1
749 RETURN

```

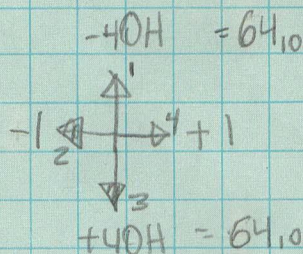
} MOVE P1

999 REM the end

$$63488_{10} = F800_{16}$$

$$64_{10} = 40_{16}$$

$$F840 = 63552$$



25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

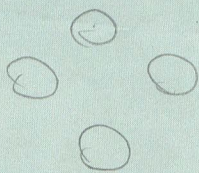
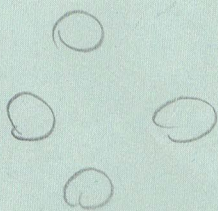
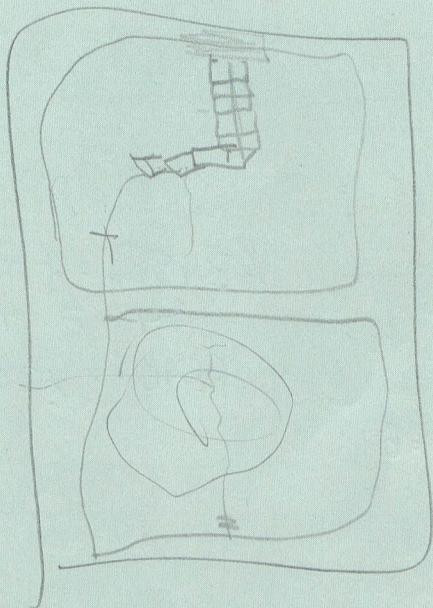
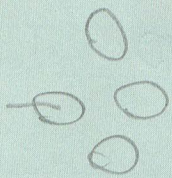
\$1,000,000,000,000,000,000.00

#####E8 is biggest for !%.C\$25F2

" E10 for !%.C\$25I

P = POSITION \ D = DIRECTION

$$\cos(\leq 102942.13)$$



BLOCKADE TWO

NON PLOTTING VERSION
EXPERIMENT NO.

- D08-3328

C08-3083

S = screen beginning / L = line length / P1, P2 = player posit
 D1, D2 = players direction / 1 = UP, 2 = left, 3 = down, 4 = Right
 P1\$, P2\$ = players symbols

```

80 S=63488 \ L=64
85 P1=64083 \ D1=4 \ P2=64060 \ D2=2
86 P1$="*" \ P2$="#"
87 !CHR$(24) → S1=ASC(P1$)+128 / S2=ASC(P2$)+128
88 INPUT "INSTRUCTIONS?" A$
89 A$=A$+" \ IF A$(1,1)="Y" THEN GOSUB 610
100 !CHR$(12) \ PLOT 0, 47, 0
110 FOR X=S TO S+L STEP L
120 POKE X, 255 \ POKE X+63, 255
130 NEXT
140 FOR X=S+1 TO S+62
150 POKE X, 255 \ POKE X+960, 255
160 NEXT
170 PLOT 1, 23, 0
180 GOTO 380
190 IF INP(0)=0 THEN GOSUB 440 ELSE 230
200 GOSUB 490
210 GOSUB 550
220 GOTO 330
230 T$=CHR$(INP(1))
240 IF T$="W" THEN P1=P1-64 \ D1=1 \ GOSUB 550 \ GOTO 330
250 " " " D " P2=P2-64 \ D2=1 \ GOSUB 490 \ GOTO 330
260 " " " A " P1=P1-1 \ D1=2 \ " 350 \ "
270 " " " O " P2=P2-1 \ D2=2 \ " 490 \ "
280 " " " Z " P1=P1+64 \ D1=3 \ " 550 \ "
290 " " " L " P2=P2+64 \ D2=3 \ " 490 \ "
300 " " " S " P1=P1+1 \ D1=4 \ " 550 \ "
310 " " " P " P2=P2+1 \ D2=4 \ " 490 \ "
320 GOTO 190 \ REM SOME OTHER LETTER WAS TYPED
330 IF PEEK(P1)=255 THEN 420
340 IF PEEK(P2)=255 THEN 430
350 IF P1=P2 THEN 410
360 IF (PEEK(P1)=S1) OR (PEEK(P1)=S2) THEN 419
370 IF (PEEK(P2)=S1) OR (PEEK(P2)=S2) THEN 429
380 POKE P1, S1 \ POKE P2, S2
400 GOTO 190

```

~~BLOCKADE 2~~
BLOCKADE 2 0-10

WIA

999 REM the end
1000 INPUT "Do you want to play again?", A\$
1010 A\$ = A\$ + " " IF A\$(1,1) <> N then 85

5 bytes per point 92	2,116 pts. versus 46x46	668 pts. 14x62	1-0 bytes per point 76
8K difference	11½K	3½K	8K difference
	PLOT	POKE	
	1/4 screen	full screen	

62
14
248
62
668

PROCSY ERROR

WHEN YOU DO 'ID'
ON SAME LINE AS 'ACCT',
'ID' is not put into I#

MINE → 13000, OHX, ELAHEP

TIM'S 10060, SED, DRMMLJ

75507, DSW, ASHAT1

MIKE H. 73457, LZ1, QWETUS

77545, OIY, PULAFI

39586, MLQ, SHYNXX

94925, DPS, TUSCAN

77507, QDZ, QUIBNE

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

0 = OK

1 = BAD ACCT

2 = ID

3 = PASS

! MORDOR NOTE

! MORDOR SANDS

! CORD ! SANDS

! NOTE

! SNOTS

! NOTE AND SNOTS

AFTER
DATASIST

PROCSY - 0-19

MASTER 1
END 2

PROCSY w/ MASTER
425 = 59 = 1

! INP (248)
249
250
251
FB?

GOES TO 4.0 now
C56
C56
OUTS DO NOTHING

A1#
I1#
P2#

HOOK IN OTHER KEYBOARD, W/O INTERRUPT. MAKE A

ROUTINE FOR USE WITH BASIC'S INP(-)

FIGURE A PROGRAM FOR EDITOR TO SWITCH BETWEEN
KEYBOARD + CASSETTE FOR INPUT

Timeshare - move RTC interrupt to higher priority vector;
hook in other keyboard at an interrupt
vector below old kybrd vector

Change appropriate ISR's, add new kybrd flag + buffer

install time share, with 2nd stack at TOP OF MEMORY

install a game or etc after timeshare

install another game etc after that

Change timeshr to also switch keyboard port for WHQ

redo a TSC program to make it reentrant

PUT TIMESHR AT 2000, 1st JOB RIGHT AFTER, 2nd
JOB AFTER THAT, AND 2ND STACK AT TOP OF
MEMORY

FIGURE OUT A PROGRAM RELOCATOR

THEN CHANGE TAPES TO DATA INSTEAD

OF ABS. BINARY, and make a relocating loader

Difficult
for 8080

ADD TCL TO PROCSY TO IMPLEMENT
PIRATE 'CREATE', 'LIST', 'RUN', AND ETC. WORK
ALSO ADD DATA FILE GET + PUT FOR PFILES
AND DATA FILE INPUT FOR ACCT, ID, PASS DATA

Data File Routines

REVISIONS ON BACK OF LAST PAGE

```
5 !CHR$(12)
10 !"Data File subsystem for the POLY-88 tape system and
11 !TAB(13), "POLY-88 BASIC version A00."
12 !TAB(16), "Written by Rob Bohn."
13 !TAB(17), FREE(0), " bytes free."
320 !\!"Commands - R = Read a file on tape"
330 !" T = Type in a data file"
340 !" E = End"
350 !\! INPUT "Command: ", C$
360 IF LEN(C$) = 0 THEN 320 ELSE C$ = C$(1,1)
370 IF C$ = "R" THEN 700
380 IF C$ = "T" THEN 400
390 IF C$ = "E" THEN !\!"The end" \ GOTO 65535
399 GOTO 320
400 !\!"Data entry section
402 GOSUB 500
401 !X$ = Input
84 DIM X$(256)
83 DIM D$(1024)
82 DIM X1$(256), X2$(256), X3$(256), X4$(256)
```


400 !\:"DATA ENTRY SECTION."

405 GOSUB 500 \: I=1

410 INPUT "FILENAME ? ", F\$

420 !\! Put cassette recorder into record mode INPUT ~

440 IF LEN(D\$) > 256 THEN GOSUB 600 \! GOTO 350

450 X\$ = FNTD\$(F\$, D\$, "500")

460 499 GOTO 350

X\$ = FNTD\$("END", "", "0")

500 IF I=1 THEN 535

505 ! CHR\$(12), TAB(15), "Data Typing Instructions"

435 R=0

535 !\! D\$="" \! "TYPE IN DATA...."

540 I = INP(2)

550 IF I=0 THEN 560 ELSE D\$ = D\$ + CHR\$(I)

555 GOTO 540

560 !\! "Data length of", LEN(D\$), " bytes."

599 RETURN

600 !\! "MULTIPLE RECORDS." \!

610 X1\$="" \ X2\$="" \ X3\$="" \ X4\$=""

611 X1\$ = D\$(1, 256)

612 IF LEN(D\$) < 512 THEN X2\$ = D\$(257) \ GOTO 650

615 X2\$ = D\$(257, 512)

620 IF LEN(D\$) < 768 THEN X3\$ = D\$(513) \ GOTO 650

625 X3\$ = D\$(513, 768)

630 IF LEN(D\$) = 768 THEN 650

640 X4\$ = D\$(769)

OVER

600's continued below

700 !\!"file input section."
710 !\INPUT "Filename? ", F\$
720 !\!"Rewind cassette recorder, connect Ext. Spkr.
730 INPUT "plug and put into PLAtmode - ", A\$
740 X\$ = FNT\$(F\$)
745 IF X\$(2,4) = "END" THEN GOTO 350
750 IF X\$(1,1) = "&" THEN !X\$(2) \ GOTO 740
760 !X\$,
799 GOTO 740
735 !CHR\$(12)

~~741 X\$ = X\$ + " " " "
~~735 !CHR\$(12)~~~~

650 !X1\$, \X\$ = FNT\$(F\$, X1\$, "500")
660 !X2\$, \X\$ = FNT\$(F\$, X2\$, "0")
670 IF LEN(X3\$) < 70 THEN !X3\$, \X\$ = FNT\$(F\$, X3\$, "0")
680 IF LEN(X4\$) < 70 THEN !X4\$, \X\$ = FNT\$(F\$, X4\$, "0")
685 X\$ = FNT\$("END", " ", "0")
690 !\RETURN

DATA FILE INPUT

```

20 DEF FNT1$(F$)
30 X=CALL(17834)\Q=18943
40 FOR X=18944 TO 19199\POKE X,0\NEXT
60 F$=F$+"8sm"\F$=F$(1,8)
70 FOR X=1 TO 8\POKE X+3155,ASC(F$(X,X))\NEXT
80 X=CALL(18864)\F1$=""
90 FOR X=3164 TO 3171\F1$=F1$+CHR$(PEEK(X))\NEXT
100 X=CALL(17834)\D$=""
120 IF F$ < F1$ THEN RETURN "&" + F1$
130 Y=PEEK(3174)\IF Y=0 THEN Y=256
140 !"**** F$, Y, bytes ****
150 FOR X=1 TO 8\ D$=D$+CHR$(PEEK(X+Q))\NEXT
160 RETURN D$\FNEND

```

TO 'USE → D\$ must be DIMENSIONED TO 256

X\$ = FNT1\$(<filename>)

→ X\$(2) = file found IF X\$(1,1) = "&" THEN GOTO ####
otherwise X\$ = Data

85 ~~X=PEEK(LEN)~~

90 ~~FOR X=3164 TO 3164~~

DATA FILE OUTPUT

NOT IMPLEMENTED

```

170 DEF FNTD$(F$,D$)
180 F$=F$+"8spa" \ F$=F$(1,8) \ Q=18943
190 FOR X=1 TO LEN(D$) \ POKE 18943+X,ASC(D$(X,X)) \ NEXT
200 X=CALL(17834) \ R=R+1 \ Q1=3179
210 FOR X=1 TO 8 \ POKE Q1+X,ASC(F$(X,X)) \ NEXT
220 POKE 3188,R
    " 3189, 0
    " " 90, LEN(D$)
    " " 91, 0
    " " 92, 74
    " " 93, 0 = ABS. BINARY, 4 = DATA
    " " 94, 0
    " " 95, 0
295 IF TIME(1) < 1200 THEN 295 ← 12G
300 CALL(18507) ! \ F$, \ X=CALL(18507)
310 RETURN F$ \ F$END
205 X=CALL(18586,33) \ X=TIME(0)

```

to use -

X\$ = FNTD\$(filename, <data>)

NOT IMPLEMENTED

12G
time
in seconds
as a string

One Record At a time Input System

Machine Code with 256 byte buffer
at 4A00-4AFF

49B0	3E 96	HEAD6: MVI A, 96H	
49B2	D3 01	OUT 1	175
49B4	CD 28 0C	CALL WHZ	145
49B7	FE E6	HEAD7: CPI 0E6H	(synch)
49B9	C2 B0 49	JNZ HEAD6	
49BC	CD 28 0C	CALL WHZ	
49BF	FE 01	CPI 1	(SOH)
49C1	C2 B7 49	JNZ HEAD7	
49C4	21 5C 0C	LXI H, RNAME	
49C7	0E 0E	MVI C, 14	
49C9	CD 94 02	CALL GET	
49CC	C0	RNZ	
49CD	21 5C 0C	LXI H, RNAME	
49D0	11 54 0C	LXI D, FNAME	
49D3	0E 08	MVI C, 8	
49D5	1A	COMP: LDAX D	
49D6	BE	CMP m	
49D7	C0	RNZ	
49D8	13	INX D	
49D9	23	INX H	
49DA	0D	DCR C	
49DB	C2 B5 49	JNZ COMP	
49DE	3A 66 0C	LDA RLEN	
49E1	4F	MOV C, A	
49E2	21 00 4A	LXI H, 4A20	BUFFER AREA
49E3			
49E8	C3 94 02	JMP GET + RETURN	

Revisions

271 IF D\$="~" THEN POKE 3193,2

741 IF (PEEK(3177)≠) AND (F=1) THEN 350

780 F=1

735 !CHR\$(12) \ F=0

~~680~~ ~~460~~ X\$=FNT\$(F\$,"~","0") 460

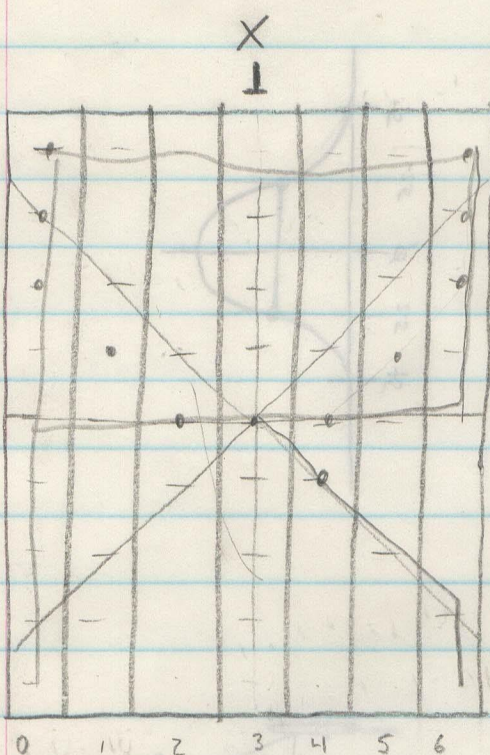
~~680~~ ~~460~~ ~~GOTO 460~~

685 X\$=FNT\$(F\$,"~","0")

X

	0-6	8-14	16-22	24-30	32-38	40-46	48-54	56-62	64-70	72-78	80-86	88-94	96-102	104-110	112-118	120-126
47																
44	(4)															
43																
35																
34	(10)															
33																
29	(5)															
28																
20	(10)															
19																
18	(5)															
14																
13	(10)															
11																
10	(4)															

7) (28) 28 (10) 19 (5) 14 (10) 13 (4) 11 10



REM ASTERISK ROUTINE
 FOR X=0 TO 6
 Y=X+1
 PLOT X,4,1
 PLOT 6-X,Y,1
 PLOT X,Y,1
 NEXT X
 FOR Y=1 TO 7
 PLOT 3,Y,1
 NEXT Y
 RETURN

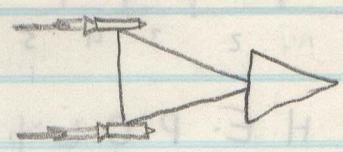
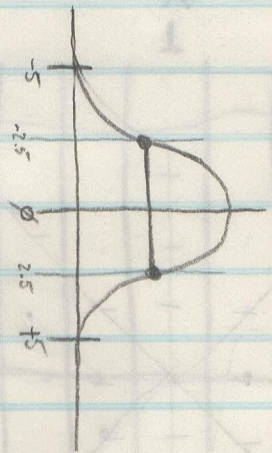
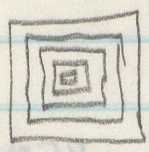
THE POLY 8 MICRUS - *

= 59

*

2

+



```

FOR X=127 TO 0 STEP -2
  FOR X=127-X1 TO X1
    FOR Y1=47 TO 0 STEP -2
      FOR Y=47-Y1 TO Y1
        PLOT X,Y,1
      NEXT Y
    NEXT Y1
  NEXT X

```

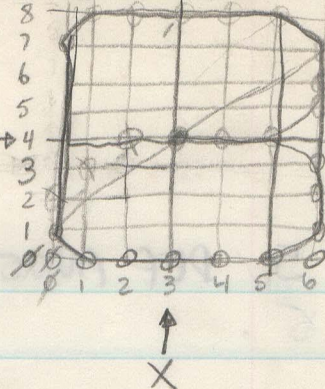
8h
821/8
91


```

DEF FNV1(X,Y)
FOR Y1=0 TO 8
PLOT X,Y1+Y,1
NEXT
RETURN 0
FNEND

```

VERTICAL LINE



① DEF FNN1(X,Y)
 Z = FNV1(X+3,Y) 1
 PLOT X+2,Y,1 \ PLOT X+4,Y,1 \ PLOT X+2,Y+7,1
 RETURN 0 \ FNEND

② DEF FNN2(X,Y)
 Z = FNH1(X,Y,7) -
 Z = FNH1(X+2,Y+4,4) -
 Z = FNH1(X+1,Y+8,5) -
 FOR Y1=1 TO 3 \ PLOT X,Y1+Y,1 \ PLOT X+6,Y1+Y+4,1 \ NEXT
 RETURN 0 \ FNEND

6,7
6,6
6,5
0,3
0,2
0,1

DEF FNH1(X,Y, Len (away from 0))

FOR XI=X TO X+L-1

PLOT XI,Y,1

NEXT \ RETURN 0 \ FNEND

HORIZONTAL LINE

③ DEF FNN3(X,Y) \ PLOT X,Y+1,1
 Z = FNH1(X+1,Y,5) \ FOR Y1=1 TO 3 \ PLOT X+6,Y1+Y,1 \ NEXT
 Z = FNH1(X+3,Y+4,3) \ FOR Y1=5 TO 7 \ PLOT X+6,Y1+Y,1 \ NEXT
 Z = FNH1(X+1,Y+8,5) \ PLOT X,Y+7,1

④ DEF FNN4(X,Y) \ Z = FNV1(X+5,Y) \ Z = FNH1(X,Y+4,7)
 FOR Y1=5 TO 8 \ PLOT X,Y1+Y,1 \ NEXT
 RETURN 0 \ FNEND

5
6
7
8 - already defined
9
0

missed 13

5 DEF FNN5(X,Y)

5

6 DEF FNN6(X,T)

6

7 DEF FNN7(X,Y)

Z = FNH1(X,Y,7)

8 FOR Y1 = 4 TO 7 \ PLOT X+6,Y1+Y,1 \ NEXT
PLOT X+5,Y1-4+Y,1

9 DEF FNN9(X,Y)

Z = FNVI(X+6,Y)

Z = FNH1(X+1,Y+4,5) \ Z = FNH1(X+1,Y+8,5) F

FOR Y1 = 5 TO 7 \ PLOT X,Y1+Y,1 \ NEXT

RETURN Ø

FNEND

Ø DEF FNNØ(X,Y)

Z = FNO(X,Y) \ PLOT X,Y,Ø \ PLOT X+6,Y,Ø \ PLOT X+6,Y+8,Ø

stack → PLOT X,Y+8,Ø \ FOR X1 = Ø TO 6 \ PLOT X1+X,X1+1+Y,1 \ NEXT

RETURN Ø

FNEND

2	2
2	2
4	2
4	2
0	0

48 ÷ 10 = 4.8

down 16 by 3 or 4 lines

* MICROCOMPUTER *

755 [V] DEF FNY(X,Y)
 FOR Y1=0 TO 4 \ PLOT X+3,Y1+Y,1 \ NEXT
 PLOT X+2,Y+4,1 \ PLOT X+4,Y+4,1
 PLOT X+1,Y+5,1 \ PLOT X+5,Y+5,1
 FOR Y1=6 TO 8 \ PLOT X,Y1+Y,1 \ PLOT X+6,Y1+Y,1 \ NEXT
 RETURN 0 \ FEND

765 [L] DEF FNL(X,Y)
 FOR X1=0 TO 6 \ PLOT X+X1,Y,1 \ NEXT
 → FOR Y=1 TO 8 \ PLOT X,Y1+Y,1 \ NEXT
 RETURN 0 \ FEND

770 [O] DEF FNO(X,Y) Z=FNE(X,Y) \ FOR Y1=1 TO 7 \ PLOT X+6,Y1+Y,1 \ NEXT
~~FOR X1=0 TO 6 \ PLOT X+X1,1 \ PLOT X+X,Y+8,1 \ NEXT~~
~~FOR Y1=1 TO 7 \ PLOT X,Y1+Y,1 \ PLOT X+6,Y1+Y,1 \ NEXT~~
 RETURN 0 \ FEND

775 [P] DEF FNP(X,Y)
 FOR X1=1 TO 5 \ PLOT X1+X,Y+8,1 \ PLOT X1+X,Y+4,1 \ NEXT
 FOR Y1=4 TO 8 \ PLOT X+6,Y1+Y,1 \ NEXT
 FOR Y1=0 TO 8 \ PLOT X,Y1+Y,1 \ NEXT
 RETURN 0 \ FEND

780 [E] DEF FNE(X,Y) \ Z=FNC(X,Y) \ FOR X1=1 TO 5 \ PLOT X1+
 X,Y+4,1 \ NEXT
~~FOR X1=0 TO 6 \ PLOT X+X1,Y,1 \ PLOT X+X1,Y+4,1~~
~~PLOT X+X1,Y+8,1 \ NEXT~~
~~FOR Y1=1 TO 7 \ PLOT X,Y1+Y,1 \ NEXT~~
 RETURN 0 \ FEND

790 [T] DEF FNT(X,Y)
 FOR X=0 TO 6 \ PLOT X+X1, Y+8, 1 \ NEXT
 FOR Y1=0 TO 7 \ PLOT X+3, Y1+Y, 1 \ NEXT
 RETURN 0 \ FEND

800 [*] DEF FNA1(X,Y)
 FOR X2=0 TO 6 \ Y2=X2+1 \ PLOT X2+X, Y+4, 1
 PLOT 6-X2+X, Y2+Y, 1 \ PLOT X+X2, Y+Y2, 1 \ NEXT
 FOR Y2=1 TO 7 \ PLOT X+3, Y2+Y, 1 \ NEXT
 RETURN 0 \ FEND

745 [8] DEF FNN8(X,Y)
 FOR X1=1 TO 5 \ PLOT X+X1, Y, 1 \ PLOT X+X1, Y+4, 1
 PLOT X+X1, Y+8, 1 \ NEXT
 FOR Y1=1 TO 7 \ PLOT X, Y1+Y, 1 \ PLOT X+6, Y1+Y, 1 \ NEXT
 PLOT X, Y+4, 0 \ PLOT X+6, Y+4, 0
 RETURN 0 \ FEND

735 [M] DEF FNM
 FOR Y1=0 TO 8 \ PLOT X, Y1+Y, 1 \ PLOT X+6, Y1+Y, 1 \ NEXT
 PLOT X+1, Y+7, 1 \ PLOT X+5, Y+7, 1
 PLOT X+2, Y+6, 1 \ PLOT X+4, Y+6, 1
 PLOT X+3, Y+5, 1
 RETURN 0 \ FEND

730 [I] DEF FNI(X,Y)
 FOR Y1=1 TO 7 \ PLOT X+3, Y1+Y, 1 \ NEXT
 FOR X1=1 TO 5 \ PLOT X1+X, Y, 1 \ PLOT X1+X, Y1+8, 1 \ NEXT
 RETURN 0 \ FEND

DEF FNL(X,Y)

FOR Y1=0 TO 8 \ PLOT X,Y1+Y,1 \ NEXT
RETURN 0 \ FNEED

PLOTS A
VERTICAL
LINE

$L = FNL(X,Y) +$
 $T = FNL(X+3,Y) +$
FOR X1=1 TO 6 \ PLOT X1+X,Y+8,1 \ NEXT

725 [C] DEF FNL(X,Y) Z=FNL(X,Y)
FOR X1=0 TO 6 \ PLOT X1+X,Y,1 \ PLOT X1+X,Y+8,1 \ NEXT
FOR Y1=1 TO 7 \ PLOT X,Y1+Y,1 \ NEXT
PLOT X+6,Y1,1 \ PLOT X+6,Y+7,1 \ RETURN 0 \ FNEED

720 [R] DEF FNR(X,Y)
Z=FNP(X,Y)

~~R = P +~~
~~E = L +~~
~~Q = E +~~
~~E = E +~~
8 = E + 1 -
~~U = L + 1~~

715 [U] DEF FNU(X,Y)
FOR Y1=1 TO 8 \ PLOT X+6,Y1+Y,1
Z=FNL(X,Y)
RETURN 0 \ FNEED

795 [S] DEF FNS(X,Y)
Z=FNE(X,Y)
FOR I=1 TO 3 \ PLOT X,Y1+Y,0 \ PLOT X+6,Y1+Y,1
NEXT \ PLOT X+6,Y+4,1
RETURN 0 \ FNEED

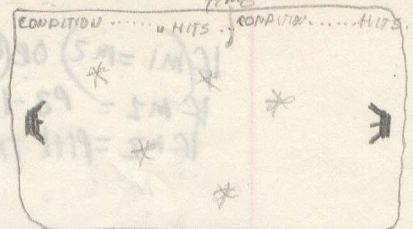
S = E - 1 + 1

A = P +
B = E +
Q = Q +
F = P -
G = L +
+

1000 of second
 move missiles update time
 see if hit update time
 update score + time update time
 blow up starship update time
 move starship if needed update time
 goto loop

63488
 1824
 64512
 63552
 +63
 63615

SPACE WAR
 GRAPH
 SPACE



016

10 CHR\$(12), \ PLOT 0,47,0 \ Z=RND(TIME(1)/65536)
 30 T1=63552 \ T2=T1+63 \ B1=B2-63 \ B2=64512+64
 40 P1=64000 \ P2=64063 (INITIAL STARSHIP POSITIONS)
 50 GOSUB 350 (PLOT STARSHIPS)

100 IF INP(0) <> 0 THEN GOSUB 200 (INPUT HANDLER)
 120 GOSUB 400 (MOVE MISSILES IF NEEDED)
 150 GOSUB 600 (update TIME + messages)
 151 GOTO 100

B1=B1-1 B2=B2-1

200 REM INPUT ROUTINE

S1=53
 S2=29
 S3=46
 S4=45

210 I\$=CHR\$(INP(1))
 220 IF I\$="W" THEN GOSUB 700 \ P1=P1-64 \ GOTO 300
 230 IF I\$="Z" THEN GOSUB 700 \ P1=P1+64 \ GOTO 300
 240 IF I\$="S" AND (M1=0) GOTO 800 (FIRE MISSILE & RETURN)
 250 " " " " " " GOSUB 750 \ P2=P2-64 \ GOTO 300
 260 " " " " " " GOSUB 750 \ P2=P2+64 \ GOTO 300
 270 " " " " " " AND (M2=0) GOTO 850 (FIRE MISSILE & RETURN)
 280 IF I\$ <> " " THEN RETURN

290 IF CHR\$(INP(1)) <> " " THEN 290 ELSE RETURN

300 OUT OF BOUNDS CHECKER

IF P1 < T1 THEN P1=T1
 IF P2 < T2 THEN P2=T2
 IF P1 > B1 THEN P1=B1
 IF P2 > B2 THEN P2=B2

MOVE ROUTINE

350 POKE P1-63, S1 \ POKE P1+1, S \ POKE P1, 0 \ POKE P1+65, S2
 360 POKE P2-65, S3 \ POKE P2-1, 40 \ POKE P2, 0 \ POKE P2+63, S4
 370 RETURN

IF $(m1 = m2)$ OR $(m1 - 1 = m2)$ OR $(m1 - 2 = m2)$ THEN 900
 IF $m1 = p2 - 1$ THEN $H2 = H2 + 1$ / 60 SUB 800
 IF $m2 = p1 + 1$ THEN $H1 = H1 + 1$ / 60 SUB 800
 POKE $m1, 63$

400 IF $m1 < 0$ THEN $m1 = m1 + 2$ IF $m1 < f1 + 61$ THEN POKE $m1, 21$
 410 IF $m2 < 0$ THEN $m2 = m2 - 2$ IF $m2 > f2 - 61$ THEN POKE $m2, 42$ RETURN
 700 POKE $p1 - 63, 63$ POKE $p1 + 1, 63$ POKE $p1, 63$ POKE $p1 + 5, 63$
 710 RETURN
 750 POKE $p2 + 5, 63$ POKE $p2 - 1, 63$ POKE $p2, 63$ POKE $p2 + 5, 63$
 760 RETURN

$f1$ $f2$
 $f1 + 63$ $f2 - 63$

800 $m1 = p1 + 2$ POKE $m1, 21$ $f1 = m1$
 RETURN

850 $m2 = p2 - 2$ $f2 = m2$ POKE $m2, 42$
 RETURN

900 POKE $m1, 0$ $AS = TIME(0)$
 POKE $m2, 0$ IF $TIME(1) < 60$ THEN 500
 POKE $m1, 63$ POKE $m2, 63$ $m1 = 0$ $m2 = 0$ RETURN

1000 IF $H1 > 4$ OR $H2 > 4$ THEN $CH = \text{Codium Yellow}$
 IF $H1 > 8$ $H2 > 8$ " " " " REP

!CH $\#1$, CH $\#2$, $H1$, "HITS", TAB(5), CH, TAB(40), $H2$, HITS
 PLOT 0, 44, 0
 RETURN

LTR LOOKER

```
30 ! " these symbols replace the corresponding screen action: "  
40 ! CHR$(138), "=" VT, " , CHR$(156), "=" FF, "  
50 ! CHR$(125), "=" CAN, " , CHR$(155), "=" DEL, " , CHR$(154), "=" TAB. "  
55 ! \!  
60 INPUT "Start, End?", S, E  
65 FOR X = S TO E  
70   Y = PEEK(X)  
80   IF Y = 9 THEN ! CHR$(154), \ GOTO 140 (TAB)  
90   " " 11 " " " (138), \ GOTO 140 (VT)  
100  " " 12 " " " (156), \ " " (FF)  
110      24      (125), \ " " (CAN)  
120      127      (155), \ " " (DEL)  
130   ! CHR$(PEEK(X)),  
140 NEXT  
150 ! \ INPUT " Do it again? ", A$ \ A$ = A$ + " "  
160 IF A$(1,1) <> "N" THEN 60  
170 ! " Y'all come back now, you hear? "
```


RANDOM WORDS

SITE, BAT,

```
30  RANDOMIZE
40  DIM A$(26)
50  A$ = "ABCDE.....WXYZ"
60  Q = INT(RND(0)*10)+1      ← WORD LENGTH
70  FOR Y = 1 TO Q
80  X = INT((RND(0)*26)+1)    ← RANDOM LETTER
90  ! A$(X,X),
100 NEXT
110 ! " ",                    ← SPACE BETWEEN WORDS
120 GOTO 60
```


RANDOM WORDS

FILE, BAT,

3D RANDOMISE

4D DIM A(5)

5D A# = "ABCDE...WXYZ"

6D Q = INT(RND(0)*10)+1

7D FOR I = 1 TO Q

8D X = INT((RND(0)*5)+1)

9D A(X,X) = A(X,X) + 1

24AC = 9388

24AC added with AC = 24AC

25AC = 9644 =

len H increased = 25AC

24AC_H = 9388

9388 added w/ A and incremented to 9644

24AC = 9388

9388 and A = 9644

2548 = 9544

~~20AC = 9013~~

A = 001₁₆

2548₁₆ = 9544
24AC₁₆ 9388
156

$CD0 = R150FF$
 $CD0 = KBUFF$

CALL MOVE INTO A ALSO

TAPE OUTPUT FUNCTION

65250

OUTPUTS 1 RECORD

65^{to}390

TAPE SYST

START 10 DEF FNT\$(F\$,D\$)

FIGURE LEN
OF DATA

MAKE FILE
NAME 8 CHAR.

TURN ON
MOTOR

MAKE
TIME=0

I second IRG

ETB

TURN OFF
MOTOR

END

```

20 L = LEN(D$) \ IF (L > 255) OR (L <= 0) THEN "LENGTH ERROR" \ RETURN ""
30 FOR X = LEN(F$) TO 8 \ F$ = F$ + " " \ NEXT \ F$ = F$(1,8)
40 X = CALL(3272)
50 X = TIME(0)
60 IF TIME(1) < 300 THEN 60
SYNC 70 FOR X = 1 TO 8 \ X = CALL(3298, 230) \ NEXT
FILENAME 80 FOR X = 1 TO 8 \ X = CALL(3298, ASC(F$(X,X))) \ NEXT
LENGTH 90 X = CALL(3298, L)
DATA 100 FOR X = 1 TO L \ X = CALL(3298, ASC(D$(X,X))) \ NEXT
110 X = CALL(3298, 23)
120 X = CALL(3294)
130 RETURN STR$(L)
140 FNEND
  
```

TRY SETTING UP, AND
THEN ASKING RBUF/TBUFF

TO USE - RETURNS LENGTH OF DATA AS A STRING

DIM D\$ and/or DATAFILE (<256)

X\$ = FNT\$(FILENAME, DATAFILE)

IF X\$ = "" THEN ERROR ELSE X = LEN OF DATAFILE

DATA TAPE FORMAT - 1 RECORD

[SYNC]	[FILENAME]	[DATA LENGTH]	[DATA]	[EOF]
8 CHAR	8 CHAR	1 CHAR	1-256 char	1 CHAR
230	ALPHABETIC	NUMERIC	ALPHABETIC	23 ASCII End transmission block

TAPEST

TAPE INPUT FUNCTION 65400

RETURNS 1 RECORD 65530

START	10	DEF FNT1\$(F\$)
MAKE F# 8 VAR	20	FOR X=LEN(F\$) TO 8 \ F\$=F\$+"_" \ NEXT \ F\$=F\$(1,8)
START MOTOR	30	X=CALL(3272) \ N\$="" \ D\$=""
LOOK FOR SYNC	40	IF CALL(3287) <> 230 THEN 40
GO TO END OF SYNC	50	IF CALL(3287) = 230 THEN 50
GET FILENAME	60	FOR X=1 TO 8 \ N\$=N\$+CHR\$(CALL(3287)) \ NEXT
FILENAME CORRECT?	70	IF F\$ <> N\$ THEN !N\$ \ GOTO 40
ROUND IT	80	!"**",F\$, "FOUND**"
GET LENGTH	90	L=CALL(3287)
GET DATA	100	FOR X=1 TO L \ D\$=D\$+CHR\$(CALL(3287)) \ NEXT
GET END	110	IF CALL(3287) <> 23 THEN !"ERROR" \ D\$(1,1)="&"
END	120	115 X=CALL(3294) RETURN D\$
	130	FNEND

70 USE - RETURNS DATA AS A STRING (UP TO 255 char)

```

DIM D$(256)
D$=FNT1$(FILENAME)
IF D$(1,1)="&" THEN ERROR ELSE D$=
    DATAFILE
    
```


E9=PCHL

DATA FILES

MACHINE CODE

DECIMAL ADDRESS	HEX ADDRESS	HEX OP CODES	MNEMONICS	REMARKS
3272	CC8	^{205/173/2} CD A0 02	CALL SETUP	
		⁶ 06	DB 06	
		¹⁷⁰ AA	DB 0AAH	
		⁶⁴ 10	DB 40H	
		²⁰⁶ CE	DB 0CEH	
		00	DB 0	
		00	DB 0	
		00	DB 0	
		^{62/33} 3E Z1	MVI A, 96H	
		^{211/1} D3 01	OUT 1	
		²⁰¹ C9	RET	
3287	CD7	^{205/40/12} CD 28 0C	CALL WH2	
		^{38/0} Z6 00	MOV H, 0	
		¹¹¹ 6F	MOV L, A	
		²⁰¹ C9	RET	
3294	CDE	¹⁷⁵ AF	XRA A	
		^{211/1} D3 01	OUT 1	
		²⁰¹ C9	RET	
3298	CE2	¹²³ 7B	MOV A	
		^{205/44/12} CD 2C 0C	CALL WH3	
		²⁰¹ C9	RET	

USE ROUTINE ON BACK
TO INSTALL
MACHINE CODE

ROUTINE TO INSTALL MACHINECODE VIA MONITOR

```

DIM A$(64), B$(45) \ A$="" \ B$=""
20 REMARK PHASE 1
30 A$="LCC8"+CHR$(13)
40 B$="CD,AD,2,6,AA,40,CE,0,0,0,3E,96,D3,1,C9,"
50 GOSUB
60 REMARK PHASE 2
70 A$="LCD7"+CHR$(13)
80 B$="CD,28,C,26,0,6F,C9,AF,D3,1,C9,CD,2C,C,C9,"
90 GOSUB 120
100 !CHR$(12), "Machine code installed." \ GOTO 160
110 REMARK INSTALLER
120 A$=A$+B$+"SPJ20BA"+CHR$(13)+"G"+CHR$(13)
130 FOR X=1 TO LEN(A$) \ OUT 0,ASC(A$(X,X)) \ NEXT X=CALL(65535)
140 A$="" \ RETURN
160 wait

```

easier way → RESTORE 65060

65000 → FOR X=0 TO 30 \ READ I \ POKE 3272+X, I

NEXT
RETURN

65060-DATA 205,173,2,6,170,64,206,0,0,0,62,150,211,1,201

70-DATA 205,40,12,8,0,111,201

80-DATA 175,211,1,201

65090-DATA 0,205,44,12,201


```

10 !\!"PROCSY VERSION 2.54"\!
15 !"INITIALIZATION PROCEDURE."
20 INPUT"TIME(24 HOUR CLOCK) (HH,MM,SS) ?","H1,M1,S1
30 Z=RND(TIME(0)/65536)\REM RANDOMIZE + SET CLOCK TO ZERO
40 INPUT"DATE (mm/dd/yy) ?" D$
50 !CHR$(12)\GOSUB 500TIME\DIM J$(23)
70 J$="JOBCARD ERROR. ILLEGAL "\READ A1,I1$,P2$
90 IF INP(1)<>2 THEN 90 (WAIT FOR CTL-B)
95 !"GOSUB 500"T1=T
100 !"TCB L001","\T$,".\",D$,".\",FULL DUPLEX."
110 INPUT"ACCOUNT?","\A$ \A=VAL(A$(1,5)) \IF A<>A1 THEN 350
120 IF LEN(A$)>5 THEN I$=A$(7) ELSE GOTO 140
140 INPUT"ID?","\I$
150 IF I$<>I1$ THEN 360
165 P1$="\ \P$="\
160 !"PASSWORD?","\
170 IF INP(0)=0 THEN 170
180 !"\","\P1$=CHR$(INP(1))
190 IF P1$=CHR$(13) THEN 210 } HALF DUPLEX INPUT
200 P$=P$+P1$ \GOTO 170
210 IF P$<>P2$ THEN 370
220 !\!"
230 !"BATCH BASIC IS DOWN. AUTOMATIC ROUTING TO ALFIE.\!"
240 INPUT"SYSTEM?","\S$
250 IF S$="" THEN 300
260 IF S$="PIRATE" THEN 300
270 IF S$="BASIC" THEN 470
280 IF S$="ALFIE" THEN 480
290 !S$," \IS CURRENTLY NOT IMPLEMENTED." \GOTO 240
300 !"PIRATE V5.4"
310 INPUT"+++",X$

```


321 IF X\$ = "TIME" THEN 700

320 IF X\$ = "LOG" THEN 380

330 !! "ERROR, UNRECOGNIZABLE COMMAND."

340 GOTO 310

350 ! J\$, "ACCOUNT NUMBER." \ GOTO 380

360 ! J\$, "ID." \ GOTO 380

370 ! \ ! J\$, "PASSWORD."

380 GOSUB 500

390 ! ! "TCB 1001.", T\$, " ", D\$, " "

400 R = (T) / 36000 ~~WRITE T (60 * 60 * 10)~~ ^{TO GET MINUTES}

410 ! "ESTIMATED SESSION COST.", %C45F2, R

420 ! "PLEASE TURN OFF TERMINAL." \ TNX.

430 ! \ ! \ GOTO 90

PIRATE
KEYWORDS

LOG, HELP, INDEX, XMIT, GET, PUT, DELETE, QXX, CREATE, NEWS

'SYNTAX

'FILENAME

'FILENAME

'FILENAME

'FILENAME

'PARAM

'FILENAME

460 DATA 77545, "OMX", "ASHAT1"

470 ! "ROUTED TO ALFIE"

480 "ALFIE" 490 GOTO 65535 65535 REM END

LINE 500 ON, SEE 2 PAGES BACK

FOR TIME SEE 2 PAGES BACK

~~TIME 500 T = INT(TIME(1)/60) + (H1*3600) + (M1*60) + S1~~

~~H = INT(T/3600) ^{H=4} 4.54361111~~

~~M = INT(T - (H*3600)) / 60 ^{m=32}~~

~~S = (T - (M*60) - H*3600) / 60 ^{3600 | 16352}~~

~~IF H ≥ 24 THEN H = H - 24 and add one to date~~

~~IF H = 0 THEN H\$ = "00:" ELSE H\$ = STR\$(H) + ":"~~

~~IF M = 0 THEN M\$ = "00:" ELSE M\$ = STR\$(M) + ":"~~

~~IF S = 0 THEN S\$ = "00." ELSE S\$ = STR\$(S) + "."~~

~~IF LEN(H\$) = 2 THEN H\$ = "0" + H\$ T\$ = H\$ + M\$ + S\$~~

~~IF LEN(M\$) = 2 THEN M\$ = "0" + M\$~~

~~IF LEN(S\$) = 2 THEN S\$ = "0" + S\$~~

* SUN * MON * TUES * WEDS * THURS * FRI * SAT *

JANUARY

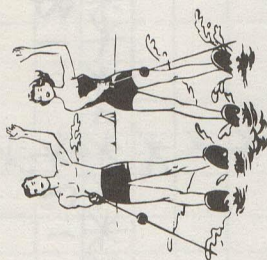
MIS * SON * MON * TUES * WEDS * THURS * FRI * SAT *

1	2	3	4			
5	6	7	8	9	10	11
12	13	14	15	16	17	18

etc

ON FABULOUS

LAKE WAWASEE



Tuesday off

207 Howard Bodenhorn

Handwritten notes on lined paper:

① 42 y
Feb 20 A R Y
MAR 20 CH
APR 20 IL
MAY 20 Y
JUN 20 E
JUL 20 Y
AUG 20 ST
SEP 20 T
OCT 20 MB
NOV 20 MB
DEC 20 MB

TEAR OFF ONLY

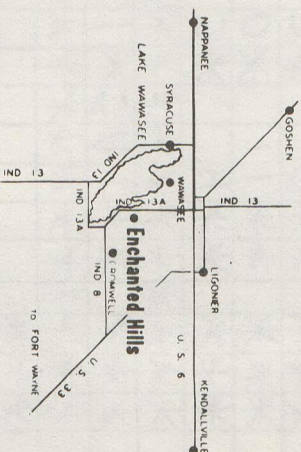
三



BY AUTOMOBILE: —
ENCHANTED HILLS is at the East end of Lake Wawasee, right on Indiana
13 Alternate, 4 miles South of its junction with U.S. 6, (see sign here) or
4 miles East of junction of Indiana 13 and 13 Alternate, (see sign here) or
2 miles North of Indiana 8 at its junction with 13 Alternate at the State
Fish Hatchery.

BANK
REFERENCE
FORT WAYNE NATIONAL BANK
FORT WAYNE, INDIANA

COLUMBIA Realty Corp. LEADERS IN RESORT DEVELOPMENT



Other BASIC pgms.

BANG (RUSSIAN ROULETTE) LISTING SAMPLE RUN - 4th page

```
10 PRINT
20 PRINT"HELLO THERE, I'M IVAN THE COMPUTER...WELCOME, YOU FOOL!!!"
30 PRINT"THIS IS A GAME OF ***RUSSIAN ROULETTE***"
40 PRINT"BY THE WAY, WHAT IS YOUR NAME?? ";
50 INPUT$
60 PRINT"OK ";A$;" , ARE YOU A CHICKEN?? (YES=1, NO=0) ";
70 INPUT$
80 IFA<>0THEN1070
90 PRINT"SO, YOU THINK YOU GOT SOME GUTS ABOUT YOU, HUH ";A$;"? GREAT."
100 PRINT"WE BOTH HAVE $100 TO, IF YOU WILL, 'BLOW'."
110 PRINT"ALSO, THIS GUN HAS SIX CHAMBERS. ";
120 PRINT"HOW MANY BULLETS DO YOU WANT?? ";
125 I=100
126 U=100
127 INPUT$
130 IFL>4THEN150
131 IFL>0THEN134
132 PRINT"YOU IDIOT ";A$;"!! HOW CAN WE PLAY WITH ONLY ";L;" BULLETS??"
133 GOTO120
134 GOTO172
135 PRINT"ONLY ONE BULLET! YOU GOT NO GUTS, ";A$;"."
136 GOTO190
137 PRINT"YOU THINK THAT YOU CAN WIN WITH TWO BULLETS, HUH ";A$;"??"
138 GOTO190
139 PRINT"THREE BULLETS, EH?? YOU LIVE DANGEROUSLY, ";A$;"."
140 GOTO190
141 PRINT"FOUR!!! I HOPE YOUR LIFE INSURANCE IS PAID, ";A$;"!!"
142 GOTO190
150 PRINT"WELL ";A$;" , IF YOU WANT ";L;" BULLETS, THEN YOU GO FIRST."
170 GOTO290
172 IFL<2THEN135
174 IFL<3THEN137
176 IFL<4THEN139
178 GOTO141
190 PRINT"OUR CHANCES ARE NOW ";6-L;" / 6 OF LIVING TO WIN!!"
200 PRINT"DO YOU WANT TO GO FIRST?? (YES=1, NO=0) ";
210 INPUT$
220 N=1-B
230 IEB<=>0THEN290
240 C$="I"
250 D$="MY"
260 E$="(RATS!!)"
262 Q=INT(RND*15)+5
270 PRINT"THEN I WILL. MY BET IS $";Q;" . WHAT'S YOURS? ";
280 GOTO670
290 C$="YOU"
300 D$="YOUR"
310 E$="(HEH, HEH)"
320 Q=INT(RND*15)+5
350 PRINT"OK ";A$;" , MY BET IS $";Q;" . WHAT'S YOURS? ";
360 GOTO670
370 PRINT"THE BET IS NOW $";D;" , WINNER TAKE ALL!!"
375 I=I-Q
380 PRINT"I NOW HAVE $";I;" LEFT, AND YOU ONLY HAVE $";U;"!!!"
390 N=N+1
400 PRINT"NOW ";C$;" SPIN THE CYLINDER, PLACE IT TO ";D$;" HEAD, ";
401 PRINT"AND PULL THE TRIGGER:"
410 X=RND
```



```

1 420 IFX<L/6THEN450
2 430 B$="CLICK"
3 440 GOTO460
4 450 B$="BANG"
5 460 PRINT
6 470 FORE=1TO65
7 480 PRINT"Z";
8 490 NEXTE
9 500 PRINTB$
10 510 IFB$<>"CLICK"THEN910
11 520 IFN/2<>INT(N/2)THEN570
12 530 C$="YOU"
13 540 D$="YOUR"
14 550 E$="(HEH, HEH)"
15 560 GOTO600
16 570 C$="I"
17 580 D$="MY"
18 590 E$="(NUTS!!)"
19 600 PRINT"WHEW, THAT WAS CLOSE!! ";D$;" TURN. ";E$
20 610 Q=INT(RND*15)+5
21 630 IFI>0THEN660
22 640 Q=I+Q
23 641 G=G+1
24 642 IFG>1THEN1065
25 650 PRINT"OH-NO!! ALL I HAVE LEFT IS $";Q;"!!!!"
26 660 PRINT"THE BET IS NOW $";D$;" I'LL ADD $";Q$;" WHAT'S YOUR RAISE?? ";
27 670 INPUTC
28 680 IFC>-1THEN750
29 690 K=K+1
30 700 IFK>1THEN730
31 710 PRINT"IF YOU PUT IN ANOTHER NEGATIVE NUMBER, I'LL KILL YOU!!!!!!"
32 720 GOTO670
33 730 PRINT"OK ";A$;" TAKE THIS: AND THIS: ---YOU'VE BEEN ZAPPED, YOU FINK!"
34 740 GOTO1090
35 750 U=U-C
36 760 IFU>=0THEN800
37 770 U=U+C
38 780 PRINT"YOU CHEATER!! YOU CAN'T BET WHAT YOU DON'T HAVE! YOU HAVE $";U
39 790 GOTO670
40 792 IFU>10THEN800
41 794 PRINT"WATCH OUT ";A$;" YOU ONLY HAVE $";U;" LEFT!!!"
42 796 GOTO820
43 800 IFC<QTHEN840
44 810 IFC>QTHEN880
45 820 D=D+Q+C
46 830 GOTO370
47 840 F=F+1
48 850 IFE>1THEN1070
49 860 PRINT"YOU CHEAPSKATE!!! MEET MY BET OR GET OUT!!"
50 865 U=U+C
51 870 GOTO670
52 880 PRINT"HEY!! WE GOT A HOTSHOT!! A BIT FOOLISH, BUT A HOTSHOT, ";
53 881 PRINT"RIGHT ";A$;"?!"
54 890 D=D+Q+C
55 900 GOTO370
56 910 PRINT"WHAT ROTTEN LUCK. ";C$;" NEEDED THAT LIKE A HOLE IN ";D$;
57 911 PRINT" HEAD. ";E$
58 920 IFN<>(2*(INT(N/2)))THEN950
59 930 C$="YOU"
60 940 GOTO960
61 950 C$="I"

```



```

960 PRINT
970 PRINT$;" WIN $";D;"!!!!!! THRILL OF THRILLS."
980 PRINT"CARE TO START OVER WITH NEW HEADS (YES=1, NO=0)";
990 INPUTG
1000 PRINT
1010 IFG<>1THEN1050
1020 PRINT"FAR OUT, ";A$;". ";
1030 D=0
1040 GOTO120
1050 PRINT"OK, IT WAS NICE PLAYING WITH YOU. BYE ";A$;". "
1060 GOTO1080
1065 PRINT"I GUESS YOU WIN, 'CAUSE I'M FLAT BROKE!! -BANG-"
1066 GOTO1080
1070 PRINT"WELL THIS IS NO CHICKENS GAME. GET LOST ";A$;"!!!!"
1080 PRINT
1090 PRINT"
1100 END

```

2:
REAR
IX
A=
B=

Ifix

NEXT PAGE HAS SAMPLE LISTING

DO 10, N=

10

DATA →
READ

SIEVE

62	6	6
8	62	9
28	69	1
98	16	7
18	66	2
68	26	8
48	6	1
38	3	1
33	3	1

SAMPLE RUN

1

DEFECTIVE

DEFECTIVE

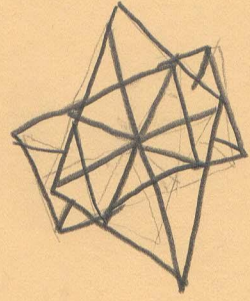
DEFECTIVE

DEFECTIVE

DEFECTIVE

DEFECTIVE

♦♦THAT'S ALL FOLKS♦♦



Figms in BASIC available

POLYMORPHIC SYSTEMS SOFTWARE USER GROUP

PolyMorphic Systems would like to encourage the interchange of software between POLY 88 users. In order to do this we will distribute Cassettes of programs we feel are of interest to other users.

Should you develop a program that you would like to see made available to fellow users, send us a cassette with a copy of the program in Byte format together with instructions (in Xeroxable form) on how to run the program. Programs we decide to distribute will be included in our price list and will be sold for 10 to 15 dollars. We accept no liability for the content or applicability of any program distributed in this manner, and offer this strictly as a service to POLY 88 users.

SOFTWARE CLASSIFICATIONS

CLASS 1: Major software fully supported by Polymorphic Systems.

This category includes software written and distributed by Polymorphic Systems with intentions for full support. "Full support" does not in any way imply that Polymorphic Systems will modify the program to meet specific user requirements. It implies only that Polymorphic Systems will take whatever steps the company feels necessary to maintain the stated functions and integrity of the program.

CLASS 2: Users group software:

This category includes software submitted by users and/or extracted from public domain. Polymorphic Systems may choose at any time to modify programs or part of a program at the company's discretion. These programs are not guaranteed to be free of errors nor will Polymorphic Systems assume any associated liability. Programs of Class 1 and 2 will be distributed by Polymorphic Systems.

CLASS 3: Unacceptable software:

This category includes software that is unacceptable, for any reason, as part of the Polymorphic Systems Program Library. Class 3 software can sometimes be changed to Class 2 if certain minimum changes are made by the person submitting the program.

PROGRAM LIBRARY

Software exchange is high on almost any list of micro-computer user's requests. To help fulfill this need, PolyMorphic Systems has instituted a Software Program Library.

This program library will consist of a broad range of user-oriented software. Programs will be contributed not only by the PolyMorphic Systems staff, but also (and in many respects more importantly) by Poly 88 users.

To encourage user-donated software, we will begin by offering a \$25.00 incentive. This one-time-only royalty will be paid to the author of the program if, and only if, the program is accepted for inclusion in the library by PolyMorphic Systems. We do not guarantee the acceptance of any particular program in advance. However, if the proper guidelines have been followed and a similar program has not previously been entered, the chances of acceptance are good.

The following guidelines are minimal requirements for any program:

- 1) Copyrighted programs will, in general, not be accepted. A possible exception is a major program submitted by you,
 - a as the copyright holder, with a written statement granting PolyMorphic Systems permission to reproduce the program on cassette tape, floppy disk, PROM, ROM, or in written form without further royalty.
- 2) Public domain programs are normally acceptable under standard guidelines if accompanied by a bibliography.
- 3) A byte standard (Poly format) cassette tape containing the program must be included.
- 4) A written explanation of the program, its applications (if not obvious), instructions for use, and your name, address and telephone number (telephone number is optional) must be included in reproducible, typewritten or neatly handwritten format on a standard PolyMorphic Systems Software Library Form

5) You must sign a statement stating that the program has been written by you or that it is truly public domain. This statement must also release reproduction rights to Polymorphic Systems.

6) The program should provide; (a) all necessary instructions and (b) sufficient documentation to allow any person of reasonable intelligence to use the program without a written explanation.

EXAMPLE: The most common display when a program is run would be a program title followed by the question "Would you like instructions (Y or N)?"

Thus the operator can obtain instructions if he does not know how the program works.

Examples often prove to be necessary for clear understanding and should be used in most programs.

7) If more than one display in succession is necessary for the instructions, the time delay between displays should be user-controlled, whenever possible.

EXAMPLE: If more than one page of instructions is necessary a statement such as "Type return to continue..." should end each page. The delay should not be a timed function controlled by the computer. This guideline is intended to allow operators with different reading speeds to proceed at their own pace.

ANOTHER EXAMPLE: Game programs should query the operator at the end of a game as to whether or not he wishes to play another game. Some operators may wish to continue immediately, while others may wish to ponder previous results before continuing. Exceptions to this rule exist, but try to follow it when possible.

EXCEPTION EXAMPLE: Dramatic effects are obtained by a timed-delay in a graphic display.

- 8) If the program expects a user input, the user should know exactly what input is expected. Any limits to numerical inputs should be given. If a program puts a question mark on the screen and halts, the software will immediately become Class 3 (see software classification list).
- 9) Long delays (while the computer is "thinking") are often confusing to an operator and some message like "thinking..." or "working..." should be displayed.
- 10) All functions claimed by the documentation must work. Exceptions should be noted.
- 11) Persons submitting a Class 2 program must assume all responsibility for supporting the program. PolyMorphic Systems only hopes to participate in the distribution of Class 2 software and will not answer any questions concerning usage. All questions will be referred to the author, who should be prepared to answer all reasonable questions accompanied by a self addressed stamped envelope.

Please do not feel we are trying to discourage user software by setting stiff requirements -- our intentions are just the opposite. The guidelines are meant only as a buffer against common problems, both for us and for you. We are always open to recommendations. If following a guideline causes more problems than benefits, let us know.

PROGRAM SUBMITTAL FORM

PROGRAM NAME _____

TYPE: ☐ Game ☐ Educational ☐ Science/
Engineering ☐ Financial
☐ Other _____

HARDWARE REQUIREMENTS _____

SOFTWARE REQUIREMENTS _____

BRIEF DESCRIPTION (25 - 100 words):

OPERATING INSTRUCTIONS:

(continue on separate sheet of paper.)

DESCRIPTION OF HOW PROGRAM WORKS:

(continue on separate sheet of paper.)

SUBMITTED BY _____ DATE _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

PHONE _____

SOFTWARE USER FEED BACK

Program Name _____ Version _____

1. How received?

- ☐ Purchased from dealer Dealer Name? _____
- ☐ By mail factory direct
- ☐ Other

2. How much memory do you have on your system? _____ K

3. How much do you expect to have?

In six months _____ K 1 year _____ K ultimately _____ K

4. How would you rate yourself as a computer user?

- ☐ Just starting
- ☐ Beginner
- ☐ Intermediate
- ☐ Wizard
- ☐ I remember the 650

5. Do you program primarily in

- ☐ BASIC ☐ Ass'y Language

About how many lines is one of your average programs? _____

6. How would you rate the time used on your machine for the following:

_____ % Games _____ % Maintenance _____ % Building programs Other _____

7. How would you compare this program to others on the market?

- ☐ Better than most
- ☐ As good as most
- ☐ Worse than most Why? _____

8. What additional programs would you like to see? _____

Comments on reverse side

I.A Software Library (BASIC)

1. Games
2. Scientific & Engineering Applications
3. Financial Analysis
4. Home Recordkeeping
5. Biorythms, Astrology, etc.
6. Programmed learning tapes (sales tool?)
7. Hobby applications (weather stations, automatic camera telescope, model railroads, navigation, etc.)
8. Industrial control (slow processes, small quantities)

I.A.I Games Classes

1. Mazes (use graphics, real time clock for blink)
2. War (star trek, tank attack)
3. Gambling (blackjack, etc.)
4. Target
5. Guessing (NIMB, etc.)
6. Economic (monopoly-like)
7. Video (LIFE, Kaleidoscope)

I.A.2 Scientific & Engineering

Engineering

1. Structural design
2. Hydraulic design
3. Electronic design
4. Aerodynamic design

Statistical

1. Confidence Tests on Data
2. Regression Model
3. Analysis of Variance
4. Probability Function Library

Mathematical

1. Matrix Operations
2. Fourier Transforms
3. Differential Equations
4. Integral Equations
5. Group Theory
6. Efficient Integration Routines

Scientific

1. Experimental design
2. Particle kinematics (Newtonian)
3. Schroedingers equation solution
4. Relativistic particle kinematics
5. Chemical binding energies
6. Entropy & thermodynamics
7. Optical design

I.A.3 Financial Analysis

1. Cash flow
2. Payroll
3. Accounts Receivable
4. Accounts Payable
5. Taxes
6. Investment return
7. Stock Market
8. Commodities
9. Mixed investments (alternate strategies)
10. Insurance as investment
11. Contract estimation
12. Inventory Control
13. Credit estimation from financial statement

I.A.4 Home Record Keeping

1. Home budgets and financial records.
2. Recipes and dietary planning.
3. Taxes (personal, ranch, small business, etc.)
4. Personal calendar
5. Personal listkeeping
6. Photograph directory
7. Valuable assets & documents directory
8. Personal investment portfolio
9. Home message center
10. Emergency information
11. Family medical records
12. Family legal, schooling, military service, welfare records, etc.

1. The first part of the report is a general
description of the project and its objectives.
2. The second part is a detailed description
of the methodology used in the study.
3. The third part is a description of the
results of the study.
4. The fourth part is a discussion of the
results and their implications.
5. The fifth part is a conclusion and
recommendations for future research.

Other Basics